# Enabling Search over Large Collections of Telugu Document Images – An Automatic Annotation Based Approach

Pramod Sankar K. and C.V. Jawahar

Centre for Visual Information Technology,
International Institute of Information Technology, Hyderabad, India
jawahar@iiit.ac.in

**Abstract.** For the first time, search is enabled over a massive collection of 21 Million word images from digitized document images. This work advances the state-of-the-art on multiple fronts: i) *Indian language* document images are made searchable by textual queries, ii) *interactive* content-level access is provided to document *images* for search and retrieval, iii) a novel *recognition-free* approach, that does not require an OCR, is adapted and validated iv) a suite of image processing and pattern classification algorithms are proposed to efficiently *automate* the process and v) the scalability of the solution is demonstrated over a *large collection* of 500 digitised books consisting of 75,000 pages.

Character recognition based approaches yield poor results for developing search engines for Indian language document images, due to the complexity of the script and the poor quality of the documents. Recognition free approaches, based on word-spotting, are not directly scalable to large collections, due to the computational complexity of matching images in the feature space. For example, if it requires 1 mSec to match two images, the retrieval of documents to a single query, from a large collection like ours, would require close to a day's time. In this paper we propose a novel automatic annotation based approach to provide textual description of document images. With a one time, offline computational effort, we are able to build a text-based retrieval system, over annotated images. This system has an interactive response time of about 0.01 second. However, we pay the price in the form of massive offline computation, which is performed on a cluster of 35 computers, for about a month. Our procedure is highly automatic, requiring minimal human intervention.

## 1 Introduction

Large collections of document images are being created from the various digitisation projects across the globe. These include the Universal Digital Library (UDL) [1], Digital Library of India (DLI) [2], Google Books, etc. [3]. Much effort is being put into the digitisation of massive quantities of documents. The popularity of these digital libraries will depend on their usability, especially through

P. Kalra and S. Peleg (Eds.): ICVGIP 2006, LNCS 4338, pp. 837–848, 2006.

content level search. For printed-document images, content level access was traditionally provided by using Optical Character Recognition (OCR) [4,5], to recognise the text. A text retrieval system would then be built over the recognised text. This approach produced satisfactory systems for the English language [6]. However, despite considerable effort, robust OCRs are not available for many Indian, Arabic and African languages. This is mostly becasue of the inherent complexity of the language owing to an extended character set, writing style and printing variations. Besides, the accuracy of OCRs reduces rapidly with degradations [5], which are common in scanned documents. The obtained text is thereby, not well-suited for indexing and retrieval.

On the other hand, recently proposed recognition free approaches, avoid explicit character recognition [7,8,9,10,11] by performing *Word Spotting* of a query in the image collection. The retrieval time using this approach is large since image matching in feature space is computationally intensive. If $N$ is the number of documents, and $M$ is the number of words in each document, then, the computations required for retrieving a single query would be of $O(N \cdot M \cdot l^2)$ ($l$ is length of feature vector for each word). If we assume that matching a pair of images requires 0.01 second, the retrieval time for each query, from a collection of 21 million images would be three days. Thus, a purely recognition-free approach is not scalable to large collections of images and queries.

The drawbacks of the previous approaches can be overcome by an *Annotation* based approach. Annotation is the process of assigning relevant keywords to a given image. With an annotated collection, an image can be represented in the text domain, enabling us to build an efficient retrieval system. Conventionally, annotation is performed by analysing a given image to identify the keywords that annotate it. *It can be observed that recognising the text using an OCR corresponds to annotating the image with the obtained text.* In this paper, we propose a novel approach called *Reverse Annotation*, where we analyse each word and find the corresponding images that it could annotate. Textual words are converted to the image domain and the generated images are matched with the words in the document. The matched documents are annotated by the textual keyword.

However, to annotate images, accurate image matching is required, which is computationally intensive. These computations need to be performed for every pair of generated and real word images. Given a vocabulary of $k$ words the order of comparisons would be $O(k \cdot N \cdot M)$. To make annotation feasible, we employ the clustering technique. In text-retrieval, clustering is used to arrange the documents in a manner that facilitates immediate retrieval. Similarly, we arrange the images such that the image matching could be performed in a hierarchy of increasing complexity and decreasing number. Images are first clustered using a coarse feature representation and a matching algorithm. These clusters are then used to index the word images for quick annotation. With this scheme, the complexity of annotation reduces to approximately $O(log(N \cdot M) \cdot logk)$.

The significance of our work is that we provide an interactive content level access to a massive collection of document images. Our approach is recognition-free, where images are accessed in the text domain through the proposed *Reverse*

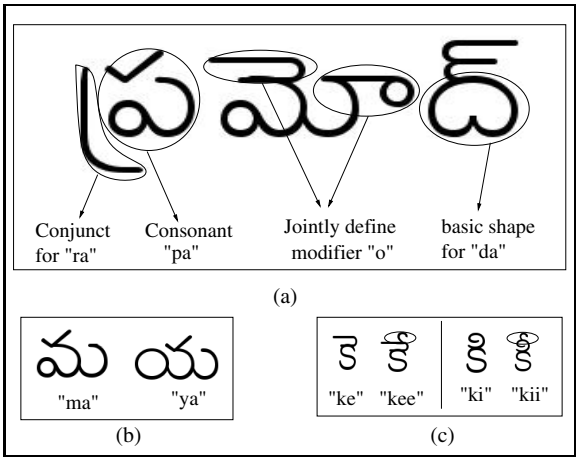



**Fig. 1.** Examples demonstrating the subtleness of the Telugu language. In (a) the consonant modifier is shown to be displaced from the consonant in different ways (b) the two characters *ma* and *ya* are distinguished only by the relative size of the circle (c) the small stroke at the top changes the vowel that modifies the consonant.

*Annotation* framework. The annotation is made computationally feasible by employing efficient clustering techniques. We demonstrate the power and scalability of our solution by creating a search engine over 500 books of Telugu language document images. The collection contained 75,000 pages with 21 million words. The search engine that was built searches the document collection in a mere 0.01 seconds.

## 2 The Challenges Faced

**Language-specific Issues.** Telugu, like most Indian languages, has a complex script, where the consonant could be modified by a vowel, consonant and/or a diacritic. A snippet of the complexity is demonstrated in Figure 1. Due to this inherent complexity of the language's script and writing style, accurate segmentation and matching of words (and characters) is a very difficult task [12].

**Issues in Scanning.** Scanned document images contain a large number of artifacts, which are cleaned on a large scale using a a semi-automatic process [3], by using various image processing operations. Owing to the variation in quality across the images, a single setup of image processing parameters would not be suitable for all. Consequently, the overall quality of the processed images is poor, thereby matching and recognising such words is very difficult.

**Scalability.** The massiveness of the digital library collections, is a serious challenge for automation of the processes. Due to this magnitude, even the quick image processing routines require large amounts of time. Despite considerable optimisations, the computation required is enormous, and the processing has

to be distributed over a cluster of computers. Managing such a cluster and transfering of large amounts of data across the network were some of the major bottlenecks in the system development.

## 3 Reverse Annotation

Content based image retrieval (CBIR) systems have thus far focused on enabling search and retrieval over relatively small image collections. With the massive increase in image collections, the scalability, performance and computational complexity issues need to be further addressed. In traditional CBIR, image matching is performed online to retrieve similar images to a given query. This online matching of queries results in large retrieval time and is thus not scalable. Indexing in the image feature space was explored in literature [13,14]. The indexing structures, such as k-d trees, are not scalable to large number of features and images. On the other hand, users are accustomed to sub-second retrieval of web pages by commercial search engines. The performance of text retrieval could be replicated for images, only by having a text-based system at this stage. This requires a textual representation for each image, which corresponds to an annotation of the images with text [15].

In the early years of image retrieval systems, images were annotated manually. For automatic annotation, the images are analysed to identify the annotation keywords, by performing image segmentation, object recognition, scene analysis etc. In recent years, cross-media relevance models have been used to annotate images based on co-occurrence of features and associated textual descriptions [16,17]. Annotations could also be learned from user feedbacks [18] or from search results over the Internet [19]. However, these techniques are not easily applicable to the domain of document images.

In our approach, instead of identifying the keywords for a given image, we identify the images that correspond to a given keyword. This scheme is called *Reverse Annotation*. In reverse annotation, we built an example image for a given keyword, and identify the images in the collection that are visually similar to it. When there is a match, the keyword is used to annotate the matched image.

This scheme is especially suitable for document images, where the knowledge of the vocabulary provides us with the possible annotations (in contrast to generic images, where annotations depend on subjectivity). For the document images, an exact keyword has to be identified for a given word image. This circumvents the problems of *synonymy* and *polysemy*, and semantic annotations which are required in the case of generic images.

### 3.1 Image Matching for Annotation

The reverse annotation problem can be stated as *"given a set of word images, identify all the word images that match a given keyword image"*. The correspondence between word- and keyword-images can be established by computing a similarity measure between each such pair. An accurate feature description and similarity measure is used for this purpose and the word is annotated with the

keyword whenever there is considerable match between the two. However, any accurate matching procedure is a computationally intensive process. If it requires about 0.05 seconds to compute the similarity between two word images, the annotation of a collection of 21 million words with a set of 30,000 keywords would require close to a thousand years. This is impractical and infeasible. To make this process feasible, we use an efficient solution derived from text retrieval, which is described in the next section.

## 4 Clustering for Annotation

In a text retrieval system, documents are indexed with the words present in them. Given a query, the documents are immediately read out of the index. It can be seen that the documents are clustered by the indexing procedure, based on the words in them. Following this strategy, we index the large collection of word images, such that similar words belong to one cluster.

At the finest level, the clusters would contain all instances of a given word in the collection, with all the variations in font type, style and size. At a coarse level, a large number of *similar-looking* words would be present in the same cluster. The feature description and similarity measure should be chosen such that they are invariant to font type, style and size changes, while being able to quickly cluster the images. Accordingly, word profile features were chosen, since they have been very useful for clustering word images [10]. The features used here are the upper word profile, lower word profile, projection profile and transition profile. The features are normalised to provide invariance to font size. Features are compared using a Dynamic Time Warping (DTW) approach since it inherently handles font type and style variations [7]. DTW is essentially a dynamic programming technique, that calculates a distance between two feature vectors, by accumulating local distances $d(i, j)$ between the $i$ th and $j$ th features of the two vectors, using the following formula:

$$D(i,j) = \min \begin{cases} D(i-1, j-1) \\ D(i-1, j) \\ D(i, j-1) \end{cases} + d(i,j)$$

**Hierarchical Clustering.** The feature representation and similarity computations between images yield non-metric pairwise distances. In such cases, the popular choice of clustering is the Hierarchical Agglomerative Clustering (HAC). HAC begins with individual clusters for each point and proceeds by merging the closest clusters until a stopping-criterion is met. However, this would require the computation of similarity between every pair of words, which is $O(N^2)$. To quicken the clustering, we only cluster those points that were not previously clustered. With such a technique, the pairwise distances need to be computed for only those words that have not yet been clustered. This results in a $O(N \cdot logN)$ algorithm, and the running time depends on the size of the clusters. With a large cluster size, we obtain coarse clusters *quickly*, since the number of points to be clustered decreases rapidly at each iteration. With smaller cluster size, the time

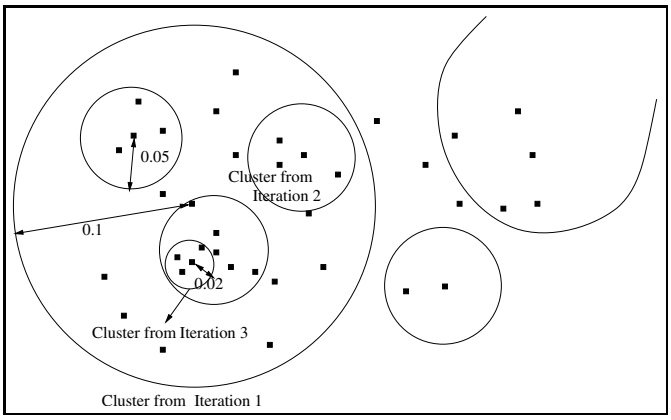



**Fig. 2.** Depiction of Clustering Procedure. In the first iteration, the data is partitioned to large clusters, quickly. Smaller clusters are then found within the larger clusters. A hierarchy of three levels of fine-ness is shown in the figure.

increases rapidly. To obtain good clusters quickly, the points are first clustered (or partitioned) coarsely and then refined to finer clusters. The assumption is that, two points cannot be found in a fine cluster, if they do not belong to a coarse cluster. The clustering is depicted in Figure 2. The cluster centroid is defined as the word with the least sum distance from the other points in the cluster.

By clustering at multiple levels, a hierarchy of clusters are built, where the number of points in the clusters reduces at each level, while the number of clusters increases. This is depicted in Figure 4. With such a hierarchy, we could identify the clusters relevant to a given keyword, and match for exact annotation within the cluster. By clustering, we eliminate a large number of comparisons which would not yield a match, thereby remarkably speeding up the annotation process. With this scheme, the number of comparisons for annotation are of $O(logk \cdot logN)$ ($K$ being number of keywords and $N$ the number of words). The annotation of 21 million words can now be performed in about 260 days (instead of the 1000 years required otherwise).

About 500 random clusters were manually evaluated to estimate the accuracy of clustering and the results are presented in Table 1.

**Table 1.** Precision-Recall of the clustering procedure, evaluated manually from 500 randomly picked clusters

| **Width of centroid word** (in pixels) | **30 - 500** | **500 - 1000** | **1000 - 1500** | **1500 - 2000** | **Total** |
|---|---|---|---|---|---|
| Precision | 92.54% | 73.91% | 73.76% | 68.53% | 72.66% |
| Recall | 62.72% | 76.69% | 80.44% | 72.39% | 75.45% |

## 5 Building the System

In this section we describe the stages involved in building the search system using the approach described above.

**Data Collection.** The data for our project was obtained from the digitisation under the Digital Library of India project. The books are available for free access at [20]. The books are digitised on a large scale at a resolution of 600dpi. Our collection consists of 500 books of the Telugu language, with 76, 425 page images.

**Segmentation.** The document images are segmented using the *docstrum* [21] algorithm. The large number of segmentation errors are corrected using the techniques described in Section 5.1.

**Feature Extraction.** Coarse features are extracted from each of the word segments. These features are the profile and transition features, which are described in Section 4.1

**Clustering.** Words are clustered using the hierarchical agglomerative clustering procedure detailed in Section 4.1. The time for clustering increases quadratically with the number of points to be clustered. To ensure that the clustering is tractable, we perform clustering over each individual book, which on an average, contains 50K words.

**Merging Clusters.** The clusters from different books are merged by comparing the cluster centroids of the respective books.

**Annotation.** The obtained clusters are annotated by finding the closest word match the cluster centroid, as elaborated in Section 5.2.

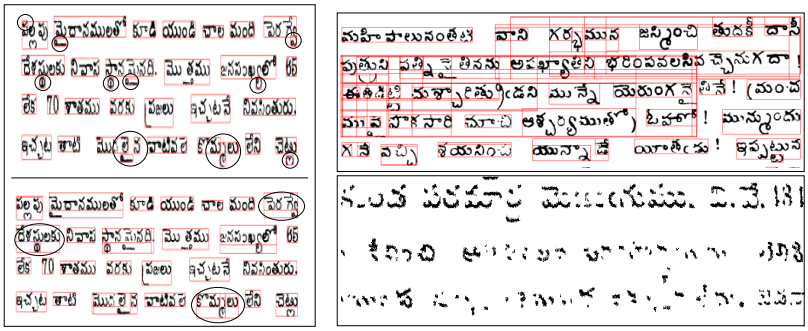
**Search Index Building.** Annotations for the clusters are used to identify the documents that correspond to each keyword. The search index is built using this correspondence. The details of the search system are described in Section 5.3.

### 5.1 Segmentation

To annotate each word, we require a segmentation of the document at word level. Due to the writing pattern of Telugu, as described in Section 2, the segmentation algorithms that work well on English documents, yield very poor results. An example is shown in Figure 3 (a), where the vowel modifiers are segmented separately from the word they belong to. In general, about 25% extra segments arise due to noise and the displaced vowel modifiers. Manual correction of these segmentation errors is infeasible, taking about three minutes per page.

The error patterns that occur in the segmentation are handled using an automatic correction scheme as

– In cases where the vowel modifier is displaced, intra-word segments occur, which generally overlap or are closer to each other than inter-word segments. The segmentation correction scheme identifies adjacent segments and merges



(a) (b)

**Fig. 3.** (a) Example of segmentation errors (above) and corrected segmentation (below). The errors are encircled and some of the corrected ones are highlighted below, (b) Over correction of poor segmentation (above), Sample page image with heavy degradations (below).

those that are closer than the average distance. An example is shown in Figure 3 (above).

- Segments from noise are considerably small in size. Accordingly all segments with dimensions less than 30 pixels are removed, which corresponds to one-twentieth of an inch, when scanned at 600 dpi.
- Segments from illustrations are generally larger than the average word size. Segments greater than 2000 pixels (three-and-half inches at 600 dpi) are, therefore, removed.

However, in some pages, due to the close proximity between successive words/ lines of text, the scheme *over-corrects*, as shown in Figure 3 (b). The outliers from incorrect segmentation, increase the computation required, but, the improvement in segmentation accuracy justifies this additional expense.

### 5.2 Annotation

For Reverse Annotation, we begin with the words of the language that are present in the document collection. These words are used to build the templates that shall be used for annotation. However, the document images do not have a parallel text. A text corpus is used to identify the words and proper nouns that are generally present in the documents of the given language. Moreover, it is well known in the information retrieval (IR) domain, that the frequency of word occurrence is roughly inversely proportional to its rank in terms of frequency, i.e., the frequency of the $k$-th most frequent word would have a frequency $f_0/k$, where $f_0$ is the frequency of the most frequent term. This is called the Zipf's law [22]. The index terms should be taken from the middle of this distribution. Highly frequent words are *stop words* and low frequency are not queried for often. With an appropriate set of words, a considerable percentage of the text and queries would be covered. Accordingly, we obtain words that are found in the

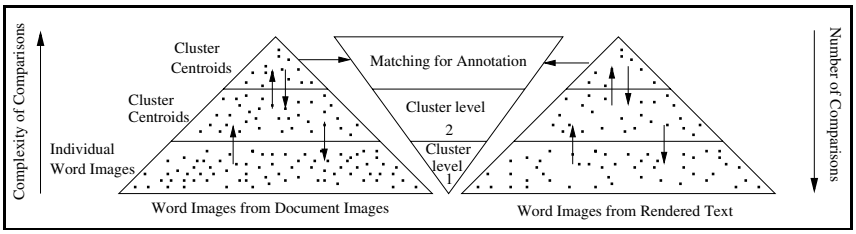



**Fig. 4.** Depiction of Annotation built above the clustering procedure

frequency range 10 to 200. The obtained set of keywords are rendered to form word images using the Eenadu font. These keyword images are also clustered using the profile features and DTW distance.

To annotate the word images, a hierarchy of comparisons are performed between the keyword- and word-images. Firstly, the cluster centroids of the word images are matched with the cluster centroids of the keyword images. The closest keyword-centroid is assigned to each word-centroid. This is performed for the two levels of hierarchy of word image and keyword clusters. We now have a correspondence between a keyword cluster and a word image cluster. An exact comparison of images can now be performed to identify the appropriate annotation for each word image. The procedure is depicted in Figure 4.

### 5.3 The Search Engine

From the word annotation, the documents that contain a given keyword can be obtained by identifying the words that are annotated by the keyword. This allows us to build the search index for the document collection. The index would contain the keywords that were used for annotation. A query is searched for in this index file and the documents containing the keyword are retrieved for the user. Since the search is in the text domain, the matching of query and index term is very quick. The system allows for querying using a transliteration scheme called *omtrans*, where the Telugu language query is entered in a Roman format. The search system has a response time of about 0.01 seconds per query. The relevant document images are retrieved for the user. Since the delivery is in the image format, the delivery of the image requires close to 3.4 seconds.

### 5.4 Computing Resources

The clustering and annotation phases require large computation resources. To make the process feasible, we distributed the computation over a cluster of 35 machines. Each machine was assigned a set of books, which were processed in a semi-automatic manner, with minimal manual intervention. One of the major challenges in this project was the handling of large amounts of data, and transferring the data across different machines.

## 6 Performance Evaluation on Ground Truth

The system built using the techniques described in this paper, was tested against a ground truth of five books, consisting of 1030 pages. The ground truth was created by manually typesetting the content of each page. The number of words in the text were 100,000, consisting of 50,000 unique words. The segmentation algorithm yielded more than 211,000 words. Following the merging of segments for segmentation-correction, the word count was 300,000. These words were clustered to 16,000 clusters. The number of words in a cluster indicates the number of words similar to the centroid in all the documents, which ideally corresponds to the word occurrence frequency in the text documents. The clusters were matched against the frequency of occurrence of each individual word in the text. The percentage of match was found to be 58.77 %.

The annotation performance was tested against the real text documents. The accuracy of annotation was calculated as the number of matching words divided by the total number of words in the given document, averaged over all documents. The accuracy of annotation was found to be 48.63%, while 24.75% of the words were annotated with a word form variation of the actual word. This is allowable, since the retrieval system would perform stemming and index a word by only its stem word. The search systems built separately over text documents and the annotated images. In case of the ground truth collection, all words were indexed, ensuring a near-perfect precision-recall. The two search engines were evaluated against 20 queries picked randomly from the keyword set. The retreival results are evaluated using the $R$-precision measure, which is the precision of the system at $R$ documents retrieved, $R$ being the number of known relevant documents for the given query in the collection. $R$ is obtained from the result of the groundtruth search system. The top 20 results were evaluated for retreival performance and the overlap in the retrieved documents was found to be about 77.38%. Thus the annotated documents are able to replciate text retrieval perfcormance to upto an accuracy of 77%. *The difference between the accuracies of the two systems comes from the inaccuracies in the image processing domain.* The errors in segmentation, clustering and annotation propogate from one stage to the next and contribute to this mismatch in the performance between purely-text based and annotated image based systems.

## 7 Related Work

Our work is similar to many of the feature indexing methods [14] and especially [7,10]. However, we annotate each of the clusters, instead of directly using them to build the index. An attempt at manual annotation of word image clusters was reported in [23], which is generally un-affordable. The motivation to use an annotation based approach comes from recent interest in automatic annotation [16,17,15]. Especially [15] uses an annotation based approach for images and videos using their textual content. Our work improves upon existing image matching systems and provides a scheme for building practical search systems for image collections.

## 8 Conclusion and Future Directions

We have demonstrated the power and effectiveness of an annotation based approach toward building search systems for document images. We tested our approach on the Indian language – Telugu, which is considered one of the most challenging to build a search system on (using conventional approaches). We built a system on 75,000 page images consisting of 21 million words, which is the largest test set used thus far in the known literature. The retrieval performance was found to be satisfactory. The approach is scalable to large collections, as is shown by our work, with the annotation time increasing linearly with the collection, while the retrieval time remains unchanged.

Since the system depends heavily on word image matching, robust and quick techniques could speed up the process. Better features and similarity measures could improve the performance of clustering, and thus of the entire system. Efficient clustering and indexing schemes could be further explored for speeding up the process. The applicability of the techniques could be tested for document images of other languages. The scalability of the approach to large digital libraries of tens of thousands of books needs to be evaluated. Finally, the results of annotation could be used to refine the segmentation of the page at word level, which could be used to learn better segmentation techniques.

## References

1. Universal Library at: http://www.ulib.org.
2. Ambati, V., N.Balakrishnan, Reddy, R., Pratha, L., Jawahar, C.V.: The digital library of india project: Process, policies and architecture. In: 2nd International Conference on Digital Libraries(ICDL). (2006)
3. Pramod Sankar, K., Vamshi Ambati, Lakshmi Pratha, Jawahar, C. V.: Digitizing a million books: Challenges for document analysis. In: 7th International Workshop on Document Analysis Systems, DAS, LNCS, Springer-Verlag (2006) 425–436
4. Mitra, M., Chaudhuri, B.B.: Information retrieval from documents: A survey. Inf. Retr. **2** (2000) 141–163
5. Doermann, D.: The indexing and retrieval of document images: A survey. In: Computer Vision and Image Understanding (CVIU) 70. (1998) 287–298
6. Taghva, K., Borsack, J., Condit, A.: Evaluation of model-based retrieval effectiveness with ocr text. ACM Trans. Inf. Syst. **14** (1996) 64–93
7. Rath, T., Manmatha, R.: Word image matching using dynamic time warping. Proc. Computer Vision and Pattern Recognition (CVPR) **2** (2003) 521–527
8. Marinai, S., Marino, E., Soda, G.: Font adaptive word indexing of modern printed documents. IEEE Trans. Pattern Anal. Mach. Intell. **28** (Aug., 2006) 1187–1199
9. Harit, G., Chaudhury, S., Ghosh, H.: Managing document images in a digital library: An ontology guided approach. In: DIAL '04: Proc. of the First International Workshop on Document Image Analysis for Libraries. (2004) 64
10. Jawahar, C.V., Million Meshesha, Balasubramanian, A.: Searching in document images. In: 4th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP). (2004) 622–627
11. Srihari, S.N., Huang, C., Srinivasan, H.: Search engine for handwritten documents. Document Recognition and Retrieval **SPIE, Vol. 5676** (2005) 66–75

12. Pal, U., Chaudhuri, B.B.: Indian script character recognition: a survey. Pattern Recognition **37** (2004) 1887–1899
13. Schmid, C., Mohr, R.: Local grayvalue invariants for image retrieval. IEEE Transactions on Pattern Analysis & Machine Intelligence **19** (1997) 530–534
14. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. ICCV. Volume 2. (2003) 1470–1477
15. Pramod Sankar K., Meshesha, M., Jawahar, C.V.: Annotation of images and videos based on textual content without OCR. In: Proc. ECCV Workshop on Computation Intensive Methods in Computer Vision. (2006)
16. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: European Conference on Computer Vision. (2002) 97–112
17. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: ACM SIGIR. (2003) 119–126
18. Wenyin, L., Dumais, S., Sun, Y., Zhang, H., Czerwinski, M., Field, B.: Semi-automatic image annotation. In: Proc. of Interact: Conference on HCI. (2001) 326–333
19. Wang, X., Zhang, L., Jing, F., Ma, W.Y.: Annosearch: Image auto-annotation by search. In: Proc. CVPR, New York, USA (June, 2006) 1483–1490
20. Digital Library of India at: http://dli.iiit.ac.in.
21. O'Gorman, L.: The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell. **15** (1993) 1162–1173
22. Zipf, G.: Human Behaviour and the Principle of Least Effort. Addison-Wesley, Cambridge, MA (1949)
23. Balasubramanian, A., Million Meshesha, Jawahar, C.V.: Retrieval from document image collections. In: 7th International Workshop on Document Analysis Systems, DAS, LNCS, Springer-Verlag (2006) 1–12